Package 'interval'

August 24, 2023

Type Package

Title Weighted Logrank Tests and NPMLE for Interval Censored Data

Version 1.1-1.0

Date 2023-08-24

Author Michael P. Fay

Maintainer Michael P. Fay <mfay@niaid.nih.gov>

Depends R (>= 2.2.1), stats, survival, perm (>= 1.0), Icens, MLEcens

Suggests coin

LazyLoad yes

Description Functions to fit nonparametric survival curves, plot them, and perform logrank or Wilcoxon type tests [see Fay and Shaw <doi:10.18637/jss.v036.i02>].

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-24 20:40:02 UTC

R topics documented:

`interval-package` *Tests and NPMLE for interval censored data*

Description

The main functions are `icfit` to fit nonparametric survival curves together with `plot.icfit` to plot them, and `ictest` to perform logrank or Wilcoxon type tests.

Details

Package:	interval
Type:	Package
Version:	1.1-0.8
Date:	2021-10-08
License:	GPL 2

Author(s)

Michael Fay

Maintainer: M.P. Fay <mfay@niaid.nih.gov>

References

Fay, MP and Shaw, PA (2010). Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R package. Journal of Statistical Software. doi:10.18637/jss.v036.i02. 36 (2):1-34.

`Aintmap` *Create A matrix and intmap*

Description

The A matrix is an n by k matrix of zeros and ones, where each row represents one of n failure times, and each column represents a possible interval for the nonparametric maximum likelihood estimate (NPMLE). The function `Aintmap` creates an A matrix and associated intmap from left and right intervals (L and R) which may not may not include the boundary of the interval (using Lin or Rin). The matrix intmap denotes the intervals of the potential jumps in the distribution of the NPMLE, and its attribute LRin denotes whether to include each of the intervals or not. Called by `icfit`.

Usage

```
Aintmap(L,R,Lin=NULL,Rin=NULL)
```

Arguments

L	numeric vector of left endpoints of censoring interval
R	numeric vector of right endpoints of censoring interval
Lin	logical vector, should L be included in the interval? (see details)
Rin	logical vector, should R be included in the interval? (see details)

Details

The Lin and Rin specify whether or not to include the ends of the intervals. They may be length 1 (and apply to all n values) or length n. The function automatically only returns the innermost intervals (also called the Turnbull intervals [see Turnbull, 1976], or the regions of the maximal cliques [see Gentleman and Vandal, 2002]). The innermost intervals give the "primary reduction" of Aragon and Eberly (1992).

Value

A list with two objects:

A	an n by k matrix of 0 and 1s
intmap	the associated intmap

References

Aragon, J and Eberly, D (1992). On convergence of convex minorant algorithms for distribution estimation with interval-censored data. J. of Computational and Graphical Statistics. 1: 129-140.

Gentleman R, and Vandal, A (2002). Nonparametric estimation of the bivariate CDF for arbitrarily censored dtaa. Canadian J of Stat 30: 557-571.

Turnbull, B (1976). The empirical distribution function with arbitrarily grouped, censored and truncated data. JRSS-B, 38: 290-295.

See Also

Called from icfit and ictest

Examples

```
Aintmap(c(2,3,3,7),c(3,5,5,8),Lin=c(FALSE,TRUE,FALSE,FALSE),Rin=c(TRUE,FALSE,TRUE,FALSE))
```

bcos *Breast Cosmesis Data*

Description

The often used data set for interval censored data, described and given in full in Finkelstein and Wolfe (1985).

Usage

```
data(bcos)
```

Format

A data frame with 94 observations on the following 3 variables.

`left` a numeric vector

`right` a numeric vector

`treatment` a factor with levels `Rad` and `RadChem`

Source

Finkelstein, D.M., and Wolfe, R.A. (1985). A semiparametric model for regression analysis of interval-censored failure time data. Biometrics 41: 731-740.

Examples

```
data(bcos)
```

getsurv *get survival values from icfit object*

Description

For a vector of times, `getsurv` gets the associated survival values. The MLE is not uniquely defined for times inbetween the first and second row on the same column of the intmap. If there is not a unique MLE for a specific time, then either use, interpolation: (default), which basically finds the point on the line connecting the two points bounding the non-unique MLE interval, or, left: take the left side of the non-unique MLE interval (smallest value) or, right: take the right side of the non-unique MLE interval. The LRin attribute is ignored (see warning).

If `icfit` has more than one strata, then performs the operations on each stratum.

Usage

```
getsurv(times, icfit, nonUMLE.method = "interpolation")
```

Arguments

times	numeric vector of times
icfit	icfit object used to define the survival function
nonUMLE.method	character vector, either "interpolation", "left" or "right". Method for finding survival when times element is not at a unique MLE time.

Value

if there is only one stratum, then creates a LIST,with elements

S	vector of survival function values at each element of times
times	vector of times for which need survival function
unique.mle	logical denoting whether associated survival value is a unique MLE
nonUMLE.method	character vector describing non-unique MLE method

if there are k>1 strata, then creates a list with k+1 elements, the elements 1:k are lists of results for each strata, and element k+1 is called strataNames and is a character vector of strata names.

Warning

The getsurv function does not use LRin attributes, so values exactly on the intmap values may only represent the limit approaching that value, not the survival at that value.

---

icfit *calculate non-parametric MLE for interval censored survival function*

Description

This function calculates the the non-parametric maximum likelihood estimate for the distribution from interval censored data using the self-consistent estimator, so the associated survival distribution generalizes the Kaplan-Meier estimate to interval censored data. Formulas using Surv are allowed similar to survfit.

Usage

```
## S3 method for class 'formula'
icfit(formula, data, ...)

## Default S3 method:
icfit(L, R,initfit =NULL, control=icfitControl(), Lin=NULL, Rin=NULL, conf.int=FALSE, ...)
```

Arguments

L	numeric vector of left endpoints of censoring interval (equivalent to first element of Surv when type='interval2', see details)
R	numeric vector of right endpoints of censoring interval (equivalent to second element of Surv function when type='interval2', see details)
initfit	an initial estimate as an object of class icfit or icsurv, or a character vector of the name of the function used to calculate the initial estimate (see details)
control	list of arguments for controling algorithm (see icfitControl)
Lin	logical vector, should L be included in the interval? (see details)
Rin	logical vector, should R be included in the interval? (see details)
formula	a formula with response a numeric vector (which assumes no censoring) or Surv object the right side of the formula may be 1 or a factor (which produces separate fits for each level).
data	an optional matrix or data frame containing the variables in the formula. By default the variables are taken from environment(formula).
conf.int	logical, estimate confidence interval? For setting conf.level, etc see icfitControl. (May take very long, see Warning)
...	values passed to other functions

Details

The icfit function fits the nonparametric maximum likelihood estimate (NPMLE) of the distribution function for interval censored data. In the default case (when Lin=Rin=NULL) we assume there are n (n=length(L)) failure times, and the ith one is in the interval between L[i] and R[i]. The default is not to include L[i] in the interval unless L[i]=R[i], and to include R[i] in the interval unless R[i]=Inf. When Lin and Rin are not NULL they describe whether to include L and R in the associated interval. If either Lin or Rin is length 1 then it is repeated n times, otherwise they should be logicals of length n.

The algorithm is basically an EM-algorithm applied to interval censored data (see Turnbull, 1976); however first we can define a set of intervals (called the Turnbull intervals) which are the only intervals where the NPMLE may change. The Turnbull intervals are also called the innermost intervals, and are the result of the primary reduction (see Aragon and Eberly, 1992). The starting distribution for the E-M algorithm is given by initfit, which may be either (1) NULL, in which case a very simple and quick starting distribution is used (see code), (2) a character vector describing a function with inputs, L,R, Lin, Rin, and A, see for example initcomputeMLE, (3) a list giving pf and intmap values, e.g., an icfit object. If option (2) is tried and results in an error then the starting distribution reverts to the one used with option (1). Convergence is defined when the maximum reduced gradient is less than epsilon (see icfitControl), and the Kuhn-Tucker conditions are approximately met, otherwise a warning will result. (see Gentleman and Geyer, 1994). There are other faster algorithms (for example see EMICM in the package Icens.

The output is of class icfit which is identical to the icsurv class of the Icens package when there is only one group for which a distribution is needed. Following that class, there is an intmap element which gives the bounds about which each drop in the NPMLE survival function can occur.

Since the classes icfit and icsurv are so closely related, one can directly use of initial (and faster) fits from the Icens package as input in initfit. Note that when using a non-null initfit,

the Lin and Rin values of the initial fit are ignored. Alternatively, one may give the name of the function used to calculate the initial fit. The function is assumed to input the transpose of the A matrix (called A in the Icens package). Options can be passed to initfit function as a list using the initfitOpts variable in icfitControl.

The advantage of the icfit function over those in Icens package is that it allows a call similar to that used in survfit of the survival package so that different groups may be plotted at the same time with similar calls.

An icfit object prints as a list (see value below). A print function prints output as a list except suppresses printing of A matrix. A summary function prints the distribution (i.e., probabilities and the intervals where those probability masses are known to reside) for each group in the icfit object. There is also a plot method, see plot.icfit.

For additional references and background see Fay and Shaw (2010).

The confidence interval method is a modified bootstrap. This can be very time consuming, see warning. The method uses a percentile bootstrap confidence interval with default B=200 replicates (see icfitControl), with modifications that prevent lower intervals of 1 and upper intervals of 0. Specifically, if there are n observations total, then at any time the largest value of the lower interval for survival is binom.test(n,n,conf.level=control()$conf.level)$conf.int[1] and analogously the upper interval bounds using binom.test(0,n). The output (CI element of returned list) gives confidence intervals just before and just after each assessment time (as defined by icfitControl$timeEpsilon).

Value

An object of class icfit (same as icsurv class, see details).

There are 4 methods for this class: plot.icfit, print.icfit, summary.icfit, and [.icfit. The last method pulls out individual fits when the right side of the formula of the icfit call was a factor.

A list with elements:

A	this is the n by k matrix of indicator functions, NULL if more than one strata, not printed by default
strata	a named numeric vector of length of pf vector (jumps in NPMLEs) for each strata, if one strata observation named NPMLE
error	this is max(d + u - n), see Gentleman and Geyer, 1994
numit	number of iterations
pf	vector of estimated probabilities of the distribution. if more than one strata, vectors are concatenated
intmap	2 by k matrix, where the ith column defines an interval corresponding to the probability, pf[i]
converge	a logical, TRUE if normal convergence
message	character text message on about convergence
anypzero	logical denoting whether any of the Turnbull intervals were set to zero
CI	if conf.int=TRUE included as a list of lists for each stratum, each one having elements time, lower, upper, confMethod, conf.level

Warning

The confidence interval method can be very time consuming because it uses a modified bootstrap and the NPMLE is recalculated for each replication. That is why the default only uses 200 bootstrap replications. A message gives a crude estimate of how long the confidence interval calculation will take (it calculates a per replication value by averaging the time of the first 10 replications), but that estimate can be off by 100 percent or more because the time to calculate each bootstrap replication is quite variable.

Author(s)

Michael P. Fay

References

Aragon, J and Eberly, D (1992). On convergence of convex minorant algorithms for distribution estimation with interval-censored data. J. of Computational and Graphical Statistics. 1: 129-140.

Fay, MP and Shaw, PA (2010). Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R package. Journal of Statistical Software. doi:10.18637/jss.v036.i02. 36 (2):1-34.

Gentleman, R. and Geyer, C.J. (1994). Maximum likelihood for interval censored data:consistency and computation. Biometrika, 81, 618-623.

Turnbull, B.W. (1976) The empirical distribution function with arbitrarily grouped, censored and truncated data. J. R. Statist. Soc. B 38, 290-295.

See Also

ictest, EMICM

Examples

```
data(bcos)
icout<-icfit(Surv(left,right,type="interval2")~treatment, data=bcos)
plot(icout)
## can pick out just one group
plot(icout[1])
```

---

icfitControl *Auxiliary for controling icfit*

Description

A function to create a list of arguments for icfit.

Usage

```
icfitControl(epsilon = 1e-06, maxit = 10000, initfitOpts=NULL, conf.level=.95, B=200,
    confMethod="modboot",seed=19439101, timeEpsilon=1e-06, timeMessage=TRUE)
```

Arguments

epsilon	The minimum error for convergence purposes. The EM algorithm stops when error $<$ epsilon, where error is the maximum of the reduced gradients (see Gentleman and Geyer, 1994)
maxit	maximum number of iterations of the EM algorithm
initfitOpts	named list of options for initfit function if initfit is function name
conf.level	level for confidence interval, used if conf.int=TRUE
B	number of bootstrap replications for conf.int=TRUE, must be at least 11
confMethod	method for confidence intervals, must be "modboot"
seed	random seed for bootstrap, if NULL no call to set.seed
timeEpsilon	small number for adding or subtracting from time for drawing confidence interval lines
timeMessage	logical, print estimate of how long modified bootstrap confidence intervals will take to calculate?

Details

There is only one option for the confMethod now. The confMethod argument is only needed for future versions if there is another confidence interval method option. For a description of the modified bootstrap method see icfit.

Value

An list with the arguments as components.

References

Gentleman, R. and Geyer, C.J. (1994). Maximum likelihood for interval censored data:consistency and computation. Biometrika, 81, 618-623.

---

ictest *do logrank or Wilcoxon type tests on interval censored data*

Description

The ictest function performs several different tests for interval censored data, and the wlr_trafo function takes interval censored data and returns one of several rank-based scores as determined by the scores option. The default for ictest is to perform a permutation test, either asymptotic or exact depending on the size of the data. Other types of tests (the scores test form or multiple imputation form) are supported. The 5 different score options allow different tests including generalizations to interval censored data of either the Wilcoxon-Mann-Whitney test (scores="wmw") or the logrank test (scores="logrank1" or scores="logrank2") (see details).

The function calls the icfit function, if an icfit object is not provided.

Usage

```
## Default S3 method:
ictest(L, R, group,
    scores = c("logrank1","logrank2","wmw","normal","general"),
    rho=NULL,
    alternative= c("two.sided", "less", "greater"),
    icFIT=NULL,
    initfit=NULL,
    icontrol=icfitControl(),
    exact=NULL,
    method=NULL,
    methodRule=methodRuleIC1,
    mcontrol=mControl(),
    Lin=NULL,
    Rin=NULL,
    dqfunc=NULL, ...)

## S3 method for class 'formula'
ictest(formula, data, subset, na.action, ...)
```

```
## Default S3 method:
wlr_trafo(x, R=NULL,
    scores = c("logrank1", "logrank2", "wmw","normal","general"),
    icFIT = NULL, initfit =NULL, control=icfitControl(),
    Lin=NULL,Rin=NULL,dqfunc=NULL,...)

## S3 method for class 'Surv'
wlr_trafo(x,...)

## S3 method for class 'data.frame'
wlr_trafo(x,...)
```

Arguments

L	numeric vector of left endpoints of censoring interval (equivalent to first element of Surv when type='interval2'), if R is NULL then represents exact failure time
R	numeric vector of right endpoints of censoring interval (equivalent to second element of Surv when type='interval2', see details)
x	response, either a Surv object or a numeric vector representing the left endpoint. If the latter and R is NULL then x is treated as exact
group	a vector denoting the group for which the test is desired. If group is a factor or character then a k-sample test is performed, where k is the number of unique

	values of group. If group is numeric then a "correlation" type test is performed. If there are only two groups, both methods give the same results.
scores	character vector defining the scores: "logrank1" (default), "logrank2", "wmw" or others (see details)
rho	either 0 (gives scores="logrank1"), or 1 (gives scores="wmw"), ignored if NULL (see Note)
alternative	character giving alternative for two-sample and trend tests, K-sample should be two.sided (see details)
icFIT	a precalculated icfit object for increased computation speed. This should be the icfit from the pooled data. Normally initfit should be used instead (see Warning)
initfit	an object of class icfit or icsurv or a character vector giving a function name, used for the initial estimate (see Warning). Ignored if icFIT is not null
icontrol	list of arguments for controling NPMLE algorithm in call to icfit (default icfitControl)
formula	a formula with response a numeric vector (which assumes no censoring) or Surv object, the right side of the formula is the group variable. No strata() is allowed
data	data frame for variables in formula
subset	an optional vector specifying a subset of observations to be used
na.action	a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action")
Surv	a Surv object, see Surv
exact	a logical value, TRUE denotes exact test, ignored if method is not NULL
method	a character value, one of 'pclt','exact.network','exact.ce','exact.mc', 'scoretest', 'wsr.HLY', 'wsr.pclt', 'wsr.mc'. If NULL method is chosen by methodRule which may use the value of exact.
methodRule	a function used to choose the method, default methodRuleIC1. (see details in perm)
mcontrol	list of arguments for controling algorithms of different methods (see mControl)
Lin	logical vector, should L be included in the interval? (see details)
Rin	logical vector, should R be included in the interval? (see details)
dqfunc	function used with general scores (see details)
control	list of arguments for controling NPMLE algorithm in call to icfit (default icfitControl)
...	values passed to other functions

Details

The censoring in the default case (when Lin=Rin=NULL) assumes there are n (n=length(L)) failure times, and the ith one is in the interval between L[i] and R[i]. The default is not to include L[i] in the interval unless L[i]=R[i], and to include R[i] in the interval unless R[i]=Inf. When Lin and Rin are not NULL they describe whether to include L and R in the associated interval. If either Lin or Rin is length 1 then it is repeated n times, otherwise they should be logicals of length n.

Three different types of scores are compared in depth in Fay (1999): When scores='logrank1' this gives the most commonly used logrank scores for right censored data, and reduces to the scores of

Sun (1996) for interval censored data. When scores='logrank2' this gives the scores associated with the grouped proportional hazards model of Finkelstein (1986). When scores='wmw' this gives the generalized Wilcoxon-Mann-Whitney scores.

The other options for scores only allow the permutation methods and follow cases where the error under the grouped continuous model is either normally distributed (scores='normal') or distributed by some other distribution (scores='general') (see Fay, 1996). For scores='general' the user must supply the function (dqfunc) which represents the density function of the inverse distribution function of the error. For example, scores='general' with dqfunc equal to `function(x){ dnorm(qnorm(x))}` gives the same results as scores='normal' or with dqfunc equal to `function(x){ dlogis(qlogis(x))}` gives the same results (theoretically, but perhaps not exactly when calculated) as scores='wmw'.

For censored data two common likelihoods are the marginal likelihood of the ranks and the likelihood with nuisance parameters for the baseline survival. Here we use the latter likelihood (as in Finkelstein, 1986, Fay, 1996, and Sun, 1996).

Because of theoretical difficulties (discussed below), the default method (method=NULL with methodRule=`methodRuleIC1`) is to perform a permutation test on the scores. There are several ways to perform the permutation test, and the function `methodRuleIC1` chooses which of these ways will be used. The choice is basically between using a permutational central limit theorem (method="pclt") or using an exact method. There are several algorithms for the exact method (see `perm`). Note that there are two exact two-sided methods and the default is to essentially double the smaller of the one-sided p-values (tsmethod='central'), while the default in the `coin` package is different (see `mControl` and the tsmethod option).

Another method is to perform a standard score test (method="scoretest"). It is difficult to prove the asymptotic validity of the standard score tests for this likelihood because the number of nuisance parameters typically grows with the sample size and often many of the parameters are equal at the nonparametric MLE, i.e., they are on the boundary of the parameter space (Fay, 1996). Specifically, when the score test is performed then an adjustment is made so that the nuisance parameters are defined based on the data and do not approach the boundary of the parameter space (see Fay, 1996). Theoretically, the score test should perform well when there are many individuals but few observation times, and its advantage in this situation is that it retains validity even when the censoring mechanism may depend on the treatment.

Another method is to use multiple imputation, or within subject resampling (method="wsr.HLY") (Huang, Lee, and Yu, 2008). This method samples interval censored observations from the nonparametric distribution, then performs the usual martingale-based variance. A different possibility is to use a permutational central limit theorem variance for each wsr (method="wsr.pclt") or use Monte Carlo replications to get an possibly exact method from each within subject resampling (method="wsr.mc").

Note that when icfit and ictest are used on right censored data, because of the method of estimating variance is different, even Sun's method does not produce exactly the standard logrank test results.

Fay and Shaw (2010) gives the mathematical expressions for the different tests.

Note that the default method performs reasonably well even when the assessment times depend on the treatment (see Fay and Shih, 2012, Fay and Hunsberger, 2013). If your primary concern is with retaining type I error and you do not mind conservativeness, then the wsr.pclt or wsr.mc methods can be used.

Value

The function `wlr_trafo` returns only the numeric vector of scores, while `ictest` returns an object of class 'ictest', which is a list with the following values.

`scores`	This is a vector the same length as L and R, containing the rank scores (i.e., the ci values in Fay, 1999 equation 2). These scores are calculated by `wlr_trafo`.
`U`	The efficient score vector. When group is a factor or character vector then each element of U has the interpretation as the weighted sum of "observed" minus "expected" deaths for the group element defined by the label of U. Thus negative values indicate better than average survival (see Fay, 1999).
`N`	number of observations in each group
`method`	full description of the test
`data.name`	description of data variables
`algorithm`	character vector giving algorithm used in calculation, value of `method` or of result of `methodRule`. One of 'pclt', 'exact.network', etc.
`statistic`	either the chi-square or Z statistic, or NULL for exact methods
`parameter`	degrees of freedom for chi-square statistic
`alternative`	alternative hypothesis
`alt.phrase`	phrase used to describe the alternative hypothesis
`p.value`	p value associated with alternative
`p.values`	vector of p-values under different alternatives
`p.conf.int`	confidence interval on p.value, for method='exact.mc' only
`nmc`	number of Monte Carlo replications, for method='exact.mc' only
`nwsr`	number of within subject resamplings, for WSR methods only
`V`	covariance matrix for U, output for method='scoretest' only
`d2L.dB2`	second derivative of log likelihood with respect to beta,output for method='scoretest' only
`d2L.dgam2`	second derivative of log likelihood with respect to gamma, output for method='scoretest' only
`d2L.dBdgam`	derivative of log likelihood with respect to beta and gamma, output for method='scoretest' only
`estimate`	output of test statistic from permutation method, difference in means in scores, output only for permutation methods
`null.value`	0, null value of test statistics from permutation method, output only with permutation methods
`np`	number of permutation replications within each WSR, for method='wsr.mc' only
`fit`	object of class 'icfit' giving results of NPMLE of all responses combined (ignoring group variable)
`call`	the matched call

Warning

Because the input of icFIT is only for saving computational time, no checks are made to determine if the icFIT is in fact the correct one. Thus you may get wrong answers with no warnings if you input the wrong icFIT object. The safer way to save computational time is to input into initfit either a precalculated icfit object or an icsurv object from a function in the Icens package such as EMICM. When this is done, you will get either the correct answer or a warning even when you input a bad guess for the initfit. Additionally, you may specify a function name for initfit. The default is NULL which uses a simple initial fit function (it is a weighted average of the A matrix, see icfit.default code). A fast but somewhat unstable function uses initcomputeMLE which uses the computeMLE function from the 'MLEcens' package. See help for icfit for details on the initfit option.

Note

The rho argument gives the scores which match the scores from the survdiff function, so that when rho=0 then scores="logrank1", and when rho=1 then scores="wmw". These scores will exactly match those used in survdiff, but the function survdiff uses an asymptotic method based on the score test to calculate p-values, while ictest uses permutation methods to calculate the p-values, so that the p-values will not match exactly. The rho argument overides the scores argument, so that if rho is not NULL then scores is ignored.

Author(s)

Michael P. Fay

References

Fay, MP (1996). "Rank invariant tests for interval censored data under the grouped continuous model". Biometrics, 52: 811-822.

Fay, MP (1999). "Comparing Several Score Tests for Interval Censored Data." Statistics in Medicine, 18: 273-285 (Correction: 1999, 18: 2681).

Fay, MP and Shaw, PA (2010). Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R package. Journal of Statistical Software. doi:10.18637/jss.v036.i02. 36 (2):1-34.

Fay, MP and Shih JH. (2012). Weighted Logrank Tests for Interval Censored Data when Assessment Times Depend on Treatment. Statistics in Medicine 31, 3760-3772.

Fay, MP and Hunsberger, SA. (2013). Practical Issues on Using Weighted Logrank Tests with Interval Censored Events in Clinical Trials. Chapter 13 in Interval-Censored Time-to-Event Data: Methods and Applications, Chen, D-G, Sun, J, and Peace, KE (editors) Chapman and Hall/CRC.

Finkelstein, DM (1986). "A proportional hazards model for interval censored failure time data" Biometrics, 42: 845-854.

Huang, J, Lee, C, Yu, Q (2008). "A generalized log-rank test for interval-censored failure time data via multiple imputation" Statistics in Medicine, 27: 3217-3226.

Sun, J (1996). "A non-parametric test for interval censored failure time data with applications to AIDS studies". Statistics in Medicine, 15: 1387-1395.

See Also

icfit, EMICM, computeMLE

Examples

```
   ## perform a logrank-type test using the permutation form of the test
data(bcos)
testresult<-ictest(Surv(left,right,type="interval2")~treatment, scores="logrank1",data=bcos)
testresult
## perform a Wilcoxon rank sum-type test
## using asymptotic permutation variance
left<-bcos$left
right<-bcos$right
trt<-bcos$treatment
## save time by using previous fit
ictest(left,right,trt, initfit=testresult$fit, method="pclt",scores="wmw")
```

---

initcomputeMLE *functions to calculate initial NPMLE of the distribution*

Description

The function icfit calculates the NPMLE of a distribution for interval censored data using an E-M algorithm with polishing and checking the Kuhn-Tucker conditions (see icfit help details). It allows functions for the initfit option in order to calculate the starting value of the distribution in the E-M algorithm. Because icfit checks the Kuhn-Tucker conditions, we can try functions without doing extensive quality control, since if the starting distribution is not close to the true NPMLE the only downside is a slower convergence. But if the initfit function is the true NPMLE then convergence happens on the first iteration. Functions must input 5 objects, L,R, Lin, Rin, and A, but need not use all of them.

Usage

```
initcomputeMLE(L,R,Lin,Rin,A=NULL,max.inner=10,max.outer=1000,tol=1e-9)
initEMICM(L=NULL,R=NULL,Lin=NULL,Rin=NULL,A=NULL,maxiter=1000,tol=1e-7)
```

Arguments

L	numeric vector of left endpoints of censoring interval (equivalent to first element of Surv when type='interval2', see icfit details)
R	numeric vector of right endpoints of censoring interval (equivalent to second element of Surv function when type='interval2', see icfit details)
Lin	logical vector, should L be included in the interval? (see icfit details)
Rin	logical vector, should R be included in the interval? (see icfit details)
A	clique matrix
max.inner	see computeMLE

max.outer	see computeMLE
tol	see either computeMLE or EMICM
maxiter	see EMICM

Details

In order to work correctly within icfit the function should output a list with at least a 'pf' element giving the estimated mass of the distribution for a series of intervals. Further, if an 'intmap' element is included (describing the series of intervals) it will be used by icfit.

Value

The function initcomputeMLE outputs an icfit object with 'pf' and 'intmap' values and some other values defined in the help for computeMLE.

The function initEMICM outputs an icsurv object with a 'pf' element but no 'intmap' element, in addition to some other values defined in the help for EMICM.

Here we define pf and intmap:

pf	vector of estimated probabilities of the distribution
intmap	2 by k matrix, where the ith column defines an interval corresponding to the probability, pf[i]

Warning

In rare cases the computeMLE function (and hence the initcomputeMLE function) can cause R to crash (at least for version 0.1-3 of the MLEcens package).

Author(s)

The wrappers for the functions were written by M. Fay, but the real work are the calculation engines:

The calculation engine for initcomputeMLE is computeMLE and was written by Marloes Maathuis, with part of the code for the optimization step is adapted from code that was written by Piet Groeneboom.

The calculation engine for initEMICM is EMICM and was written by Alain Vandal and Robert Gentleman

See Also

icfit, computeMLE, EMICM

Examples

```
## If you want speed and trust the MLEcens package, then there is no need to use icfit at all
## (but the convergence checks in icfit do not take much additional time)
data(bcos)
fit<-initcomputeMLE(bcos$left,bcos$right)
summary(fit)
plot(fit)
```

mControl *Auxiliary for feeding parameters to different methods*

Description

A function to create a list of arguments for ictest.

Usage

```
mControl(cm=NULL,nmc=10^3-1,seed=1234321,digits=12,p.conf.level=.99,
    setSEED=TRUE,tol.svd=10^-8,nwsr=10^3-1,np=10^3-1,tsmethod="central")
```

Arguments

cm	a choose(n,m) by n matrix, used if method='exact.ce', ignored otherwise
nmc	number of Monte Carlo replications, used if method='exact.mc', ignored otherwise
seed	value used in set.seed if method='exact.mc', or any of three wsr methods, ignored otherwise
setSEED	logical, set to FALSE when performing simulations
p.conf.level	confidence level for p value estimate, used if method='exact.mc', ignored otherwise
digits	number of digits to use in signif for precision of test statistics
tol.svd	tolerance for use in calculating g-inverse, values less than tol.svd are set to zero, used when method='scoretest'
nwsr	number of within subject resamples, used when method='wsr.mc', 'wsr.HLY', or 'wsr.pclt'
np	number of permutation replications within each wsr, used when method='wsr.mc'
tsmethod	two-sided method for exact permutation tests, either 'central' or 'abs' (see permControl)

Details

When cm=NULL the resulting matrix is created by chooseMatrix, it may be optionally provided here only so that chooseMatrix does not need to be repeatedly called in simulations. Also when doing simulations (with method='exact.mc' or any of the wsr methods), use setSEED=FALSE so that the seed is not reset to the same value each time you call the function.

See calcPvalsMC for description of how p.conf.level is used.

Value

An list with the arguments as components.

`methodRuleIC1` *Rule for determining method for ictest*

Description

This is the default function which determines which permutation method (e.g., 'pclt' or 'exact.network') to use in `ictest`.

Usage

```
methodRuleIC1(x, group, exact, Nbound = c(20))
```

Arguments

`x`	vector of response scores
`group`	group membership vector
`exact`	logical, TRUE=exact method chosen, FALSE=pclt
`Nbound`	bound, if n>Nbound then method='pclt' otherwise either 'exact.mc' (for k-sample or trend) or 'exact.network' (for two-sample)

Details

This function determines which of several methods will be used in `ictest`, see `permTS` for description of methods.

When exact=FALSE then returns 'pclt'. When exact=TRUE then returns either 'exact.network' if the length(cc)<=Nbound and it is a two-sample test or 'exact.mc' otherwise. When exact=NULL and the length(cc)<=Nbound, then returns either 'exact.network' (for two-sample) or 'exact.mc' (for k-sample and trend). When exact=NULL and length(cc)>Nbound returns 'pclt'.

Value

a character vector with one of the following values: "pclt","exact.network","exact.mc"

See Also

`ictest`

plot.icfit *Plot icfit object*

Description

Plots either the survival distributions, the cumulative distributions, or a transformation of the cumulative distributions, from an `icfit` object. If there is more than one strata, all strata will be plotted. Note that for interval censored data, the changes in the NPMLE of the survival function usually do not occur at unique points but occur within some interval where any of an infinite number of curves will maximize the likelihood. We show those intervals were the NPMLE is indeterminate as a gray rectangle.

Usage

```
## S3 method for class 'icfit'
plot(x,XLAB="time",YLAB=NULL,COL=gray((8:1)*.1),LTY=1:9,LEGEND=NULL,
    XLEG=NULL,YLEG=NULL,shade=TRUE,dtype="survival",
    dlink=function(x){log(-log(1-x))}, xscale=1, yscale=1, conf.int=NULL,
    estpar=list( lty=NULL, lwd=1, col=gray(0)),
    cipar=list( lty=1:9, lwd=1, col=gray(0.8)),
    ...)
```

Arguments

x	an icfit object, see icfit
XLAB	x label
YLAB	y label, if NULL label matches dtype
COL	a vector representing color of rectangles of indeterminate NPMLE, COL[i] used for ith strata
LTY	a vector for lty values for lines, LTY[i] used for ith strata
LEGEND	logical value, include legend or not, if NULL set to TRUE only if number of strata>1
XLEG	x location for legend, if NULL then gives maximum of 0 and minimum time from intmap
YLEG	y location for legend
shade	logical, should the rectangles of indeterminate NPMLE be colored?
dtype	type of distribution plotted, one of 'survival', 'cdf' or 'link' (see details)
dlink	link function when dtype='link' (see details)
xscale	a numeric value used to multiply the labels on the x axis. So if the data are in days, then a value of 1/365.25 would give labels in years.
yscale	a numeric value used to multiply the labels on the y axis. A value of 100, for instance, would be used to give a percent scale. As in the survival package, only the labels are changed, not the actual plot coordinates

conf.int	logical, should confidence intervals be plotted? NULL plots them if they are present in x object and gives no errors if they are not.
estpar	list of par arguments for the estimated distribution lines. If lty=NULL uses LTY argument, otherwise ignores LTY (for backward compatability)
cipar	list of par arguments for the confidence interval lines
...	other arguments passed to the plot function

Details

Turnbull (1976) noted that the NPMLE was not unique within a certain set of intervals. We represent that non-uniqueness using colored rectangles when shade=TRUE. The option shade=TRUE is not supported when dtype="link".

The option dtype="cdf" plots the cumulative distribution function.

When there are several strata, different types of weighted logrank-type tests (see ictest) may be derived from score statistics under the grouped continuous model with error distribution known. To test which test is appropriate, one may plot the cumulative distribution for each stratum transformed by the inverse of the proposed error distribution (see Fay, 1996). These are plotted with dtype="link" where dlink is the link function which transforms the cdf. The "wmw" scores correspond to dlink=qlogis, the "logrank2" scores correspond to the default complementary log-log dlink, and the "normal" scores correspond to dlink=qnorm.

Value

Returns a list of arguments for the legend. Values are x,y, legend, fill, lty. See legend help.

Note

An object of class 'icsurv' from the Icens package can use this plot function by redefining its class to 'icfit' and 'plot.icfit' will work on it.

References

Fay, MP (1996). Rank invariant tests for interval censored data under the grouped continuous model. Biometrics. 52: 811-822.

Turnbull, B.W. (1976) The empirical distribution function with arbitrarily grouped, censored and truncated data. J. R. Statist. Soc. B 38, 290-295.

See Also

icfit

Examples

```
data(bcos)
fit1<-icfit(Surv(left,right,type="interval2")~treatment,data=bcos)
summary(fit1)
plot(fit1)
```

summary.icfit *Methods for icfit objects*

Description

The print method prints as a list, except the A (clique) matrix. The summary method prints the masses an associated maps for the fit. The [method allows picking out of specific fits for individual elements of the factor when the right hand side of the formula in `icfit` was a factor.

Usage

```
## S3 method for class 'icfit'
summary(object, digits=4,  ...)

## S3 method for class 'icfit'
print(x, ...)

## S3 method for class 'icfit'
x[i]
```

Arguments

`object`	an icfit object
`x`	an icfit object
`digits`	number of digits for rounding results
`i`	scalar integer to pick ith strata
`...`	arguments to be passed

See Also

`ictest`

Examples

```
data(bcos)
icout<-icfit(Surv(left,right,type="interval2")~treatment, data=bcos)
print(icout)
summary(icout)
icout[1]
```

SurvLR *Transform Surv object to data frame with L and R values*

Description

Takes a Surv object and transforms it into a data frame with two variables, L and R, representing the left and right interval of interval censored data. The failure time is known to be in the interval (L,R]. Right censored data are handled by setting L=R for observed and R=Inf for right censored. These are interpreted correctly by icfit and ictest.

Usage

```
SurvLR(x)
```

Arguments

x	a Surv object

Details

Currently type='counting' not supported.

Value

A data frame with two variables:

L	left end of interval
R	right end of interval

See Also

Called from icfit and ictest

Examples

```
time<-c(1,5,3,7)
status<-c(1,1,0,1)
y<-Surv(time,status)
SurvLR(y)
```

Index